Syllabus

Computational Mathematics

Contact Information:
Prof. Sivan Leviyang
Sivan.Leviyang@georgetown.edu
St. Mary's Hall. Office 318

Textbook: There is no textbook. I will provide papers and book excerpts for reading. I will also post class lectures as pdf and mp4 files.

Office Hours: TBD. See the discord server.

Grading:
Weekly Homeworks : 65%.
Final : 35%.

Homework will be posted on Dropbox (link available on discord) Friday and will be due the next Thursday in class as a hard copy. I encourage you to work together on homeworks! I will create a discord channel for every homework. Feel free to post questions, solutions, code, tips, etc.

Coding is an essential component of this course and every homework will involve some coding. Please code in Python. If you want to code in another language, please talk to me.

Canvas: All homeworks and course documents will be posted on Dropbox with links available on discord. I do not use Canvas.

Course Content: We will focus on computational methods in regression, classification, and dimension reduction (i.e. machine learning).

1. Finite Dimensional Linear Methods
2. Infinite Dimensional Linear Methods
3. Methods Assuming Structure
4. Nonlinear Methods

MATH 4536 Weekly Schedule of Topics and Assignments

I, Tristan Devictor, created this approximate schedule based on lecture notes.

Week 1: Syllabus, computation basics

Week 2: Linear algebra review (rank, spectral theorem, projection, matrix multiplication shortcuts), homework 1

Week 3: Principal component analysis, orthogonal matrices, homework 2

Week 4: Singular value decomposition, homework 3

Week 5: Independent component analysis, Eckhart-Young-Mirsky Theorem, homework 4

Week 6: QR decomposition, algorithm efficiency analysis, numerical methods for computing eigenvalues (e.g., power iterations), homework 5

Week 7: Linear regression (least-squares), homework 6

Week 8: Hilbert spaces, Kernel methods, homework 7

Week 9: Gradient descent, Newton's method, logistic regression, log-likelihood function, homework 8

Week 10: Reproducing kernels, homework 9

Week 11: Karush-Kuhn-Tucker conditions, homework 10

Week 12: Primal and dual functions for optimization, Lloyd's algorithm, homework 11

Week 13: Support vector machines, penalized least squares, Bayes optimal classifiers, homework 12

Week 14: Spectral clustering, graph theory, homework 13

Week 15: Neural networks, backpropagation, autoencoders, homework 14

Week 16: Neural networks continued

Week 17: Final exam review, final exam

Homework #1

1. Let u and v be two column vectors with dimension n. Letting $A = uv^T$, what is the dimension of A. What is A_{ij} in terms of the coordinates of u and v?

2. Let x be an n dimensional vector, and M be a $n \times n$ dimensional matrix. Show that $x^T Mx = \sum_{i=1}^n \sum_{j=1}^n M_{ij} x_i x_j$. (Hint: Start by noticing that the kth coordinate of Mx, $(Mx)_k$, is given by $(Mx)_k = \sum_{j=1}^n M_{kj} x_j$)

3. Let b and x be vectors in $\mathbb{R}^n$, Show that $\nabla(b^T x) = b$.

4. Let A be an $n \times n$ matrix and $x \in \mathbb{R}^n$. Show that $\nabla(x^T Ax) = (A + A^T)x$. Hint: Use problem 2.

5. Let $x \in \mathbb{R}^n$ and $f(x) = x^T Ax + b^T x + c$ where A is a symmetric $n \times n$ matrix, b is a n dimensional vector and c is a scalar. Assume A is inverticle and solve for the critical point of $f(x)$. Under what conditions is the critical point a maximum, minimum? What can you say about critical points if A is not invertible? (Justify your statements.)

Homework #2

1. Let V be an $n \times k$ orthonormal matrix. (Recall, this means the columns of V are orthonormal vectors.)

   (a) Show that $V^T V = I$ where I is the $k \times k$ identity matrix.

   (b) Show that VV^T is a projection matrix.

   (c) Show that if $n > k$ then VV^T is not the identity matrix and if $n = k$ then $VV^T = I$ where I is the $n \times n$ identity matrix.

   (d) Let $x \in \mathbb{R}^k$. Show $\|Vx\| = \|x\|$.

2. Let P be a symmetric, projection matrix in $\mathbb{R}^n$. Show that $P = VV^T$ for some orthonormal matrix V. This result is a converse to 1b. (Hint: Consider the linear space $\mathcal{S} = \{x \mid Px = x\}$ and its orthogonal complement $\mathcal{S}^\perp$. If you construct V as a basis for $\mathcal{S}$ and use the symmetry of P to show that $Py = 0$ for $y \in \mathcal{S}^\perp$ then you'll be able to verify $P = VV^T$.)

3. Let M be an $p \times p$ matrix. M is positive definite if $x^T M x > 0$ for all $x \in \mathbb{R}^p$ and $x \neq 0$.

   (a) Let M be a symmetric matrix. Use the spectral theorem to show that M is positive definite if and only if all its eigenvalues are positive.

   (b) Suppose $M = X^T X$ for some matrix X. Show that M is positive definite as long as the columns of X are linearly independent.

4. Attached you will find the senators dataset. There are 100 rows corresponding to 100 senators. The first column contains senator names. The rest of the columns contain how the senator voted with $1, -1, 0$ meaning for, against, and abstain.

   (a) In class we derived the formula for computing a projection onto a linaer space as well as showing that the principle components should be the eigenvectors of the covariance matrix. Write out a derivation of these results.

   (b) Compute a 2-dimensional PCA of the data restricted to the vote columns. Before applying the PCA, center each column by subtracting off its mean. Form a scatter plot the scores, which will be two dimensional, with the color of the points given according to the senator's party.

   (c) PCA is an unsupervised dimension reduction, meaning that we don't have any labeling associated with the samples. In this case, we do know the senator's party affiliations. Let v be the vector given by the difference between the mean Republican vote vector and the mean Democrat vote vector. Perform a 1-d dimension reduction by projecting onto v and compare to a 1-d PCA dimension reduction.

Homework #3

1. Let X be an $n \times p$ matrix. Below you'll prove the svd $X = USV^T$ with the number of columns in U and V will be restricted to $\min(n, p)$.

   (a) Assume $n \leq p$. Prove the svd. (To do this construct the v_i from X^TX, but keep just n of them. Then you'll be able to define $s_i u_i = Xv_i$ for $i = 1, 2, \ldots, n$ and the rest of the proof will follow.)

   (b) Assume $n > p$. Prove the svd. (This time use all the v_i from X^TX. Construct $s_i u_i = Xv_i$ for $i = 1, 2, \ldots, p$ and that's all you'll need for the rest of the proof.)

2. Here are some odds and ends to prove.

   (a) Let A be $n_1 \times p$ and B be $n_2 \times p$. Show,

   $$AB^T = \sum_{i=1}^{p} a_i b_i^T \tag{1}$$

   where a_i and b_i are the ith columns of A, B respectively.

   (b) Let X be $n \times n$ and symmetric. Prove the spectral decomposition $X = QDQ^T$ and $X = \sum_{i=1} \lambda_i q_i q_i^T$. (Here the q_i are the eigenvectors of X and λ_i are the eigenvalues. Q is the matrix with the q_i as columns and D is diagonal with $D_{ii} = \lambda_i$. Hint: show $X = QDQ^T$ on the basis of the q_i.)

   (c) (Not sure if you've seen this before.) Assume X is symmetric and positive semidefinite (i.e. $v^TXv \geq 0$ for all $v \neq 0$.). Let $X = QDQ^T$ be the spectral decomposition of X. Show that $M = \sqrt{D}Q^T$ satisfies $X = M^TM$ where $\sqrt{D}$ is D with its diagonal elements replaced by their square root. Loosely, we can view M as the square root of X. Is M unique? Show that there is no square root if X is not positive semidefinite.

3. Let X be $n \times p$. Show that $\|X\|_F^2 = \sum_{i=1}^{\min(n,p)} s_i^2$ where s_i is the ith singular value of X. (Hint: Use the svd $X = \sum s_i u_i v_i^T$ and $\|X\|_F^2 = \text{trace}(X^TX)$.)

4. Let A be an $n \times p$ matrix. All columns of A have one entry equal to 1 and all other entries equal to 0. Let $\sum_{j=1}^{p} A_{ij} = \tau_i$ for $i = 1, 2, \ldots, n$. What are the singular values of A? What are the left singular vectors? What can you say about the right singular vectors? (Hint: AA^T has a simple form. That'll give you information about the svd of A^T.)

5. Go back to your code for the senators data from last weeks homework. Rewrite your computations by computing the svd of the data matrix X rather than using the covariance matrix X^TX.

Homework #4

1. In this problem, you'll prove the shortened version of Eckhart-Young that we went over in class. Let X be an $n \times p$ matrix with svd decomposition $X = \sum_{i=1}^{\min(n,p)} s_i u_i v_i^T$. Let $B = \sum_{i=1}^{k} s_i u_i v_i^T$. and assume $s_k > s_{k+1}$ for a given k. We want to determine a solution to the minimization,

$$\min_{Y \sim n \times p, \text{rank}(Y)=k} \|X - Y\|_2^2. \tag{1}$$

   (a) Show $\|X - B\|_2^2 = s_{k+1}^2$.

   (b) Suppose Y is rank k. Show that Y must have the form,

$$Y = \sum_{i=1}^{k} r_i a_i b_i^T, \tag{2}$$

   where $r_i \in \mathbb{R}$, $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}^p$.

   (c) Show that there exists $w \in \text{span}(v_1, v_2, \ldots, v_k, v_{k+1})$ with w orthogonal to every vector in $\text{span}(b_1, b_2, \ldots, b_k)$. Then show $\|X - Y\|_2^2 \geq s_{k+1}^2$. Conclude that B is a solution to (1).

2. Consider again the MNIST dataset. Let X be the data matrix. Let x_i be the ith row of X, corresponding to the ith image. For all values of k, determine the fraction of the dataset's variance captured by a k-dimensional PCA. For a given k-dimensional PCA, let $x_i' \in \mathbb{R}^{784}$ be the PCA approximation (i.e. after decoding) of x_i. Consider image 1, which is a 5. Use different k to compute x_1' and visualize the projection using matplotlib's imshow function. For which k do you begin to see the 5? How does that relate to the fraction of variation captured by the PCA? (For this problem do not use sklearn's PCA. You may use numpy or scipy's eigenvalue or svd functions.)

3. This problem will set the scene for implementing ICA on the next homework. In this problem we will construct S and A matrices that we will then use to construct X. Recall $X \sim n \times p$, $A \sim n \times k$, and $S \sim k \times p$. In the next homework, we will pretend we only know X and see how well ICA reconstructs S and A.

   (a) Write a function sample_signal(p) that return a random vector $s \in \mathbb{R}^p$ as follows. s_1 should be chosen randomly from the values $\{1, 2, 3, 4\}$ with each value having equal probability. Then iteratively construct the s_i for $i = 2, 3, \ldots, p$ as follows. With probability 0.9, $s_i = s_{i-1}$ and with probability 0.1, s_i is chosen from $\{1, 2, 3, 4\}$ with each value having equal probability.

   (b) Use sample_signal(p) with $p = 100$ to sample 5 signals and row bind them to form the S matrix which will be $5 \times p$. Plot the 5 signals, the rows of S, just to see their difference. Let $n = 100$. Decide on a sensible way to define the A matrix that linearly combines the S matrix to from the data matrix X. Add some noise to the data by defining,

$$X = AS + 0.1\eta, \tag{3}$$

   where η is an $n \times p$ matrix with each entry given by a independent standard normals. (The point of adding η is that it will make X rank n. Without it X has the rank of S, which is 5, and in the steps below M will not be invertible. We'll discuss this more fully in class.)

(c) Write a function that performs the following two transforms on X and outputs the matrix Z:

i. Let Y be the row centered version of X for which all rows have mean zero. (**Careful, in PCA we center the columns, but in ICA we center the rows**. I forgot to emphsize that in class.)

ii. Let $\Sigma = \frac{1}{p} YY^T$. Note Σ is $n \times n$. Construct an invertible $n \times n$ matrix M such that $\Sigma = MM^T$. (M is often written as $\Sigma^{1/2}$.). Let $Z = M^{-1}Y$.

(d) Let $w \in \mathbb{R}^n$ with $\|w\| = 1$. Let $y = w^T Z$. Note that $y \in \mathbb{R}^p$. Show

i. $\frac{1}{p}\sum_{i=1}^{p} y_i = 0$

ii. $\frac{1}{p}\sum_{i=1}^{p} y_i^2 = 1$

(e) Let

$$G(x) = -\exp[-\frac{x^2}{2}] \tag{4}$$

If you've taken probability do part (i), if not do part (ii),

i. Let X be a standard normal r.v. Compute $E[G(X)]$. Let $x_1, x_2, \ldots, x_p$ be iid samples from X. Explain why $E[G(x)]$ is a good estimator of $\frac{1}{p}\sum_{i=1}^{p} G(x_i)$.

ii. Determine,

$$\int_{-\infty}^{\infty} f(x)G(x)dx, \tag{5}$$

where $f(x) = \frac{1}{\sqrt{2\pi}}\exp[-\frac{x^2}{2}]$. Use numpy or scipy's normal sampler to sample 1 million standard normals (i.e. a normal with mean 0 and variance 1). Compare the value of the integral to $\frac{1}{p}\sum_{i=1}^{p} G(x_i)$ where x_i is the ith of your 1 million samples.

Homework # 5

1. Read Lecture 10 of Trefethon and Bau (see the Course Documents-Books folder) on Householder reflections.

   (a) Let $\|v\| = 1$, show that $I - 2vv^T$ is an orthogonal matrix.

   (b) Let A be an $n \times p$ matrix with $n > p$ and assume that A is full rank (i.e. the columns are linearly independent). Derive the complexity of using Householder reflection to from the QR decomposition of A. (Trefethon and Bau do this in the Lecture, but try first to do it on your own.)

   (c) Code a function `QR_householder(A)` that forms the QR decomposition of the matrix A using Householder reflection. Besides standard python, your function should only make use of numpy's matrix-vector arithmetic. Show some examples demonstrating that your function is correct.

2. In Lecture 4 of Trefethon and Bau, do exercise 4.1.

3. In Lecture 7 of Trefethon and Bau, do the following exercises.

   - 7.2
   - 7.3 (recall that det(AB)=det(A)det(B) and use the QR decomposition)
   - 7.4 (the $P^{(1)}$ and $P^{(2)}$ of the problem are the spans of the given vectors)
   - 7.5 (Note that you cannot assume that the QR is formed by Gramm-Schmidt.)

   The QR we discussed in class is Trefethon's reduced rank QR. For the reduced rank QR, if $X = QR$ then X and Q have the same dimensions. Trefethon distinguishes the full rank QR in which Q is extended to be square.

4. Attached you will find a file `X.mtx`. The mtx file format is used to store sparse matrices. Use scipy's sparse library to load and manipulate the sparse matrix. For some orientation, see the attached `sparse.py`.

(a) Use a power iteration to compute the first two dominant eigenvalues/vectors of XX^T. Use only matrix-vector arithmetic. You can compare to the correct values as computed by `scipy.sparse.linalg.eigsh(mm, k=2)`. `eigsh` uses a Lanczos iteration which we will discuss next week.

(b) You can convert X to a dense matrix by

```
Xdense = X.toarray()
```

`Xdense` is a numpy matrix. Try to repeat (a) but using `Xdense`.

Homework # 6

1. Create for yourself a dataset containing $x_1, x_2, \dots, x_n \in \mathbb{R}$ and $y_1, y_2, \dots, y_n \in \mathbb{R}$ with $n = 1000$. Choose the x_i randomly with uniform distribution on the interval $[0, 10]$ and set $y_i = 1 + 2x_i + \epsilon_i$ where ϵ_i is chosen from a standard normal. Now given the data, consider the least squares problem,

$$\min_{\alpha \in \mathbb{R}^2} \sum_{i=1}^{n} |y_i - \alpha_1 - \alpha_2 x_i|^2 \tag{1}$$

   (a) Let X be an $n \times 2$ matrix with $X_{i1} = 1$ and $X_{i2} = x_i$ for $i = 1, 2, \dots, n$. Show that the solution to the minimization is given by $\alpha = (X^T X)^{-1} X^T y$.

   (b) Compute the optimal α and plot the line $y = \alpha_1 + \alpha_2 x$ and the data points (x_i, y_i) on one graph to show the fit.

2. This is essentially problem 11.2a in Trefethon and Bau. Suppose we would like to approximate the function $f(x) = 1/x$ on the interval $[1, 2]$ using a linear combination of the functions $\sin(x)$, e^x and the gamma function $\Gamma(x)$ by finding $\alpha \in \mathbb{R}^3$ that minimizes,

$$\int_1^2 (f(x) - (\alpha_1 \sin(x) - \alpha_2 \mathrm{e}^x - \alpha_3 \Gamma(x))^2 \mathrm{d}x \tag{2}$$

   Write code that estimates the answer using a discretization of $[1, 2]$ and a least squares problem. Given your estimate of the optimal α, create a plot comparing $f(x)$ to the approximating linear combination. See

   `https://en.wikipedia.org/wiki/Gamma_function`

   for a definition of $\Gamma(x)$. The library scipy.special has a gamma function.

3. From homework 5, consider again the sparse matrix X of problem 4. Use a random matrix approach to compute the first 2 singular values of X. Experiment with different values for the q. You may also find that by trying to compute more than 2 singular values, you get a more accurate estimate of the first 2. You can check your answer using eigh or your orthogonalized power iteration from homework 5.

Homework # 7

1. The Courant-Fischer theorem expresses eigenvalues in variational form, i.e. as an optimization problem. Let A be an $n \times n$, symmetric matrix. Let $\lambda_k(A)$ be the kth eigenvalue of A, ordered from greatest to least. Courant-Fischer states,

   $$\lambda_k(A) = \max_{dim(S)=k} \min_{x \in S, x \neq 0} \frac{x^T A x}{x^T x} \tag{1}$$

   and,

   $$\lambda_k(A) = \min_{dim(S)=n+1-k} \max_{x \in S, x \neq 0} \frac{x^T A x}{x^T x} \tag{2}$$

   Prove both forms for the cases $k = 1, k = 2$. The key insights come from the spectral theorem. (I hope the proof of the general case is then clear, but you don't need to go through it.)

2. Let X be an $n \times p$ matrix. Show that $\|X\|_2 = s_1$ where s_1 is the dominant singular value and $\|X\|$ is the spectral norm.

3. Let $z_1, z_2, z_3, \ldots, z_n \in \mathcal{H}$ where $\mathcal{H}$ is a Hilbert space. (If you haven't studied Hilbert spaces, you can assume $\mathcal{H} = \mathbb{R}^p$.) Let $V' = \text{span}(v_1, v_2, \ldots, v_k)$ for $v_i \in \mathcal{H}$ solve the PCA problem,

   $$\min_V \sum_{i=1}^{n} \|z_i - \text{proj}_V(z_i)\|_{\mathcal{H}}^2, \tag{3}$$

   where $\text{proj}_V(z_i) = \text{argmin}_{w \in V} \|w - z_i\|_{\mathcal{H}}^2$. ($\|\cdot\|_{\mathcal{H}}$ is the norm of $\mathcal{H}$, if you are assuming $\mathcal{H} = \mathbb{R}^q$ it's just the Euclidean norm.) Let $Z = \text{span}(z_1, z_2, \ldots, z_n)$. The goal of this problem is to show $V \subseteq Z$. (We did this in class. Here I want you to go through it yourself and fill in the details.)

   (a) Show that all $x \in \mathcal{H}$ have a unique decomposition,

   $$x = x^Z + x^\perp, \tag{4}$$

   where $x^Z \in Z$ and $x^\perp$ is orthogonal to every element of Z.

   (b) Let $V^* = \text{span}(v_1^Z, v_2^Z, \ldots, v_k^Z)$. Show

   $$\sum_{i=1}^{n} \|z_i - \text{proj}_{V'}(z_i)\|_{\mathcal{H}}^2 \geq \sum_{i=1}^{n} \|z_i - \text{proj}_{V^*}(z_i)\|_{\mathcal{H}}^2, \tag{5}$$

(c) Show that $V' \subseteq Z$ excluding the trivial case of all $z_i = 0$.

4. Let $x_1, x_2, \ldots x_n \in \mathbb{R}^p$ and $y_1, y_2, \ldots, y_n \in \mathbb{R}$. Let α' be a solution of the least squares problem,

$$\min_{\alpha \in \mathbb{R}^p} \sum_{i=1}^{n} \|y_i - x_i^T \alpha\|_2^2. \tag{6}$$

(a) Show that we can assume $\alpha' \in \text{span}(x_1, x_2, \ldots, x_n)$.

(b) Let K be a $n \times n$ matrix with $K_{ij} = x_i^T x_j$. Let $\alpha' = \sum_{i=1}^{n} \beta_i x_i$. Show that we can determine the β_i using only K and the y values but no other information about the x_i.

Homework # 8

Due 3/13

1. Let $f(x,y) = 100(y-x^2)^2 + (1-x)^2$. This is the famed banana function (see wikipedia). The minimum of f is $(1,1)$ (why?). Here we know the minimum, so we don't need to numerically optimize, but the banana function serves as a simple setting to compare gradient descent to Newton's method.

   (a) Starting at the point $(4,4)$ use a gradient descent with a fixed learning rate to locate the minimum. How many iterations before you find the minimum? You can alter the learning rate to get the quickest convergence you can. You can also vary the learning rate over the iterations.

   (b) Now repeat but use Newton's method. How many iterations now?

2. Here we'll use a logistic regression to build a classifier for the MNIST dataset. To keep things manageable, let's build a classifier that determines whether an image is a 3 or not a 3. (If we have a chance in subsequent homeworks, we'll build a classifier for all the digits.) For each MNIST image x_i, let $y_i = 1$ if the ith image is a 3, otherwise $y_i = 0$. The $x_i \in \mathbb{R}^{784}$, but add a leading 1 to each image to make $x_i \in \mathbb{R}^{785}$. (This will allow the linear function $\alpha \cdot x$ below to have a non-zero intercept term.) Recall the logistic regression model,

$$P(y=1|x,\alpha) = \frac{1}{1+\exp(-\alpha \cdot x)}, \tag{1}$$

   where $x \in \mathbb{R}^{785}$ is an MNIST image. Define the likelihood function,

$$L(\alpha) = \prod_{i=1}^{n} P(y_i \mid x_i, \alpha), \tag{2}$$

   where n is the number of images.

   (a) Show that $\ell(\alpha) = \log L(\alpha)$ satisfies,

$$\ell(\alpha) = \sum_{i=1}^{N} (1-y_i)(-\alpha \cdot x_i) - \log(1+\exp(-\alpha \cdot x_i))) \tag{3}$$

(b) Derive a formula for the gradient and Hessian of $\ell(\alpha)$. Show that the Hessian is semi-negative definite (i.e. all eigenvalues are non-positive).

(c) Split the MNIST dataset into a training dataset and a test dataset. Using the training dataset, apply steepest ascent to find the α that maximizes $\ell(\alpha)$. Then use Newton's method on the training dataset to find the optimal α. You will find that the Hessian may not be invertible. In this case consider perturbing the Hessian to $-\rho I + H$ where $\rho > 0$. Explain why this will make the Hessian invertible. Use this perturbed matrix instead of the Hessian in Newton's method.

(d) Given the optimal α that you computed above, construct a classifier that determines if an image is a 3 or not. Apply your classifier to the test dataset and determine your accuracy.

Homework # 9

1. Attached you will find a dataset `dataset.csv` containing $n = 1000$ samples (x_i, y_i) with $x_i \in \mathbb{R}^2$ and $y_i \in \{0, 1\}$.

   (a) Visualize the dataset by generating a scatter plot of the x_i and using the value of y_i to color the points.

   (b) Apply a logistic regression to the datasets. Recall, we assume
   $$P(y = 1 \mid x, \alpha) = \frac{1}{1 + \exp[-\alpha \cdot x]} \tag{1}$$
   and then find α that maximizes the log-likelihood. As in a previous homework, add a leading 1 to each x_i so that your fit of the logistic regression gives an optimal $\alpha \in \mathbb{R}^3$.

   (c) Use the fit from (b) to generate a classifier $\Phi(x) : \mathbb{R}^2 \to \{0, 1\}$. Letting α^* be the optimal α from (b),
   $$\Phi(x) = \begin{cases} 1 & \text{if } P(y = 1 \mid x, \alpha^*) \geq .5 \\ 0 & \text{if } P(y = 1 \mid x, \alpha^*) < .5 \end{cases} \tag{2}$$
   Plot the decision boundary of your classifier. The decision boundary will be a line that separate points $\Phi(x) = 1$ from $\Phi(x) = 0$. Add the data points x_i on top. What is the accuracy of your classifier?

2. Consider the polynomial reproducing kernel $k(x, y) = (x^T y + 1)^d$ for $d \in \mathbb{N}$ and $x, y \in \mathbb{R}^2$. Let H be the function space generated by $k(x, y)$.
   $$H = \text{span}\{f(x) : \mathbb{R}^2 \to \mathbb{R} \mid f(x) = k(x, y) \text{ for some } y \in \mathbb{R}^2\}. \tag{3}$$
   Show that H is the space of all degree d polynomials. Write down a basis for H. What is the dimension of H?

3. Continuing with the notation of problems 1 and 2, let
   $$z_i = k(x, x_i) \tag{4}$$
   and consider the dataset (z_i, y_i) for $i = 1, 2, 3, \ldots, n$. (Note $z_i \in H$.) Define the logistic regression model on H by,
   $$P(y = 1 \mid z, \alpha) = \frac{1}{1 + \exp[- < \alpha, z >]}, \tag{5}$$

for $\alpha \in H$. $<,>$ is the inner product of H defined by k. To fit the logistic regression, we need to solve the following optimization.

$$\max_{\alpha \in H} \sum_{i=1}^{n} y_i \log P(y_i \mid z_i, \alpha) + (1 - y_i) \log(1 - P(y_i \mid z_i, \alpha)). \quad (6)$$

Let α^* solve (6).

(a) Show that we can assume $\alpha^* = \sum_{i=1}^{n} \beta_i z_i$.

(b) Rewrite (6) as an optimization in terms of β. Letting $K_{ij} = k(x_i, x_j)$ show that all you need to determine β are the y_i and the matrix K. Using either steepest ascent or Newton's method, find the optimal β for $d = 5$ and $d = 50$.

(c) Let β^* be the optimal β you found in (b). Given β^*, define the classifier $\Psi(z) : H \to \{0, 1\}$ analogous to $\Phi(x)$ in problem 1. We can use Ψ to define a classifier $\tilde{\Psi} : \mathbb{R}^2 \to \mathbb{R}$, For $w \in \mathbb{R}^2$,

$$\tilde{\Psi}(w) = \Psi(k(x, w)). \quad (7)$$

Compute the decision boundary of $\tilde{\Psi}$. One simple way to do this is to put a grid down on $\mathbb{R}^2$. At the grid points evaluate $\tilde{\Psi}$. This will give you a general idea of the decision boundary. Show the boundary for the case $d = 5$ and $d = 50$.

Homework # 10

1. For $\alpha > 2$ and $0 < a \leq 1$, solve

$$\min_{x \in \mathbb{R}^p} \sum_{i=1}^{p} x_i^{\alpha} \tag{1}$$
$$\text{subject to } \|x\|^2 = 1,$$
$$x_i \geq a \text{ for } i = 1, 2, \ldots, p.$$

2. (This is a well known problem in econometrics.) Let x_t be the units of cake we eat on day t for $t = 1, \ldots, T$. We require $x_t \geq 0$ for each t and $x_1 + x_1 + \cdots + x_T = 1$. (We always eat a non-negative amount of cake and we have a total of 1 unit of cake to eat.) Define,

$$f(x) = \sum_{t=1}^{T} \beta^{t-1} u(x_t), \tag{2}$$

where $\beta \in (0, 1)$ and $u(w)$ is the amount of utility we derive from eating w units of cake. $f(x)$ models the total utility we derive from eating the cake. Use the KKT conditions to find x that maximizes utility for $u(w)$ given by,

(a) $u(w) = \sqrt{w}$,

(b) $u(w) = w^2$.

(BE CAREFUL, the goal is to MAXIMIZE the utility.)

3. Let $y \in \mathbb{R}^n$ and A an $n \times p$ matrix. Consider the constrained optimization

$$\min_{x \in \mathbb{R}^p} \|y - Ax\|^2 \tag{3}$$
$$\text{subject to } \sum_{i=1}^{p} x_i^2 \leq a^2$$

(a) Use the KKT conditions to solve this optimization.

(b) Consider the ridge regression,

$$\min_{x \in \mathbb{R}^p} \|y - Ax\|^2 + \rho\|x\|^2, \tag{4}$$

for $\rho > 0$. Let x^* be the solution. Show $x^* = (A^T A + \rho I)^{-1} A^T y$. Be sure to show that $(A^T A + \rho I)^{-1}$ always exists.

(c) Assume $(A^T A)^{-1}$ exists. Show that for any $\rho > 0$, if $a = \|x^*\|$ then the constrained optimization is also solved by x^* with Lagrange multiplier equal to ρ. (This provides a $1-1$ correspondence between ridge regression and the constrained optimization in (a). The assumption of $(A^T A)^{-1}$ is not needed and we'll remove it in subsequent work.)

4. Consider the constrained optimization,

$$\begin{aligned} &\min_{x \in \mathbb{R}^p} \|y - Ax\|^2 \\ &\text{subject to } \; x_i \geq 0 \text{ for } i = 1, 2, \ldots, p \end{aligned} \tag{5}$$

Write code that implements a projected gradient approach for solving this problem. Construct and test your code for various y, A combinations. (Don't forget to monitor the loss function!)

Homework # 11

1. Attached is a review of non-negative matrix factorization. Read sections 1, 2, and 3.1, paying particular attention to the optimization algorithms described in section 3.1 In next week's homework, we will implement an NMF algorithm.

2. Consider the convex optimization

$$\min_{x\in\mathbb{R}^n} \|x\|^2 \tag{1}$$

$$\text{subject to: } a^T x = b \tag{2}$$

$$x \geq 0 \tag{3}$$

   where $a \in \mathbb{R}^n$.

   (a) Find the solution of the primal problem. (Hint: As a simplified case assume that all coordinates of $a_i > 0$.)

   (b) Determine the dual problem.

3. Let $y \in \mathbb{R}^p$. Consider

$$\min_{x\in\mathbb{R}^p} \|y - x\|_2^2 \tag{4}$$

$$\text{subject to:} \|x\|_2^2 \leq K \tag{5}$$

$$\tag{6}$$

   Assume that $\|y\|_2^2 > K$.

   (a) Show the problem is convex.

   (b) Show that the optimal x is tiven by,

$$x^* = \sqrt{K}\frac{y}{\|y\|_2} \tag{7}$$

   (c) Write down the dual problem and solve it. Relate the solution of the dual problem to the Lagrange multiplier of the solution to the primal.

4. For a dataset $x_i \in \mathbb{R}^p$ with $i = 1, 2, ..., n$ consider the k-means problem,

$$\min_{a,\mu_1,\mu_2,\ldots,\mu_k} \sum_{i=1}^{n} \|x_i - \mu_{a_i}\|^2 \tag{8}$$

   (Above, I'm using the same notation as discussed in lecture.)

(a) Implement Lloyd's algorithm as a Python function. (Write your own implementation.)

(b) Show that Lloyd's algorithm is a descent algorithm.

(c) Apply your code from (a) to the MNIST dataset with $k = 10$. Produce an image showing the μ_j for $j = 1, 2, ..., 10$. How homogeneous are the clusters in terms of the image numbers they contain?

Homework #12

1. Consider the least squares problem,

$$\min_{x \in \mathbb{R}^p} \|y - Ax\|_2^2, \tag{1}$$

where A is an $n \times p$ matrix and $y \in \mathbb{R}^n$. Let x^* be a solution of this problem. (We do not assume that x^* is a unique solution.)

(a) Consider the problem,

$$\begin{aligned} &\min_{x \in \mathbb{R}^p} \|x\|^2 \\ &\text{subject to } \|y - Ax\|_2^2 = \|y - Ax^*\|^2. \end{aligned} \tag{2}$$

Let $A = USV^T$ be the svd decomposition of A. Let s_i be the ith singular value of A, so that $S_{ii} = s_i$. Let S^+ be the $p \times n$ diagonal matrix defined by,

$$S_{ii}^+ = \left\{ \begin{array}{ll} \frac{1}{s_i} & \text{if } s_i \neq 0 \\ 0 & \text{if } s_i = 0 \end{array} \right. \tag{3}$$

Show that the solution x^+ of (2) satisfies $x^+ = VS^+U^Ty$. The matrix US^+V^T is called the pseudoinverse of A. (Hint: You won't need KKT here. Instead, expand x in the V basis and y in the U basis. You'll be able to determine Ax and derive an expression for $\|y - Ax\|^2$. Then you'll be able to show that x^+ satisfies the constraint and has minimal norm.)

(b) Consider the penalized least squares problem,

$$\min_{x \in \mathbb{R}^p} \|y - Ax\|_2^2 + \lambda \|x\|^2. \tag{4}$$

with $\lambda \geq 0$. Let x_λ be the solution to the penalized problem. We know that for $\lambda > 0$, the solution is unique. Show that $\|x_\lambda\|$ is decreasing in λ and that $\lim_{\lambda \to 0} x_\lambda = x^+$. (Hint: You know $x_\lambda = (A^TA + \lambda I)^{-1}A^Ty$. Plug in the svd for A, write y in the U basis and then show $x_\lambda \cdot v_i = x^+ \cdot v_i$.

(c) Go back to homework 10 and redo 3(a) using x^+. You should be able to determine whether the Lagrange multiplier satisfies $\nu = 0$ based on $\|x^+\|$. (In my solutions to hw

10, I had a lemma which noted $\|x_\lambda\| \to x_0$ or $\|x_\lambda\| \to \infty$ as $\lambda \to 0$. This is true, with $x_0 = x^+$, but you can now show that the $\|x_\lambda\| \to \infty$ case doesn't happen.)

(d) Go back to homework 10 and redo $3(c)$, but do not assume the existence of $(A^T A)^{-1}$.

2. The file `dataset_separable.csv` provides separable data (x_i, y_i) for $i = 1, 2, \ldots, 1000$. $x_i \in \mathbb{R}^2$ and $y_i \in \{-1, 1\}$. Recall the SVM for separable data.

$$\begin{aligned} &\min_{a \in \mathbb{R}^2, b \in \mathbb{R}} \|a\|^2 \\ &\text{subject to } y_i(a^T x_i + b) \geq 1. \end{aligned}$$

(a) Derive the dual problem. (I did this in class. I want you to go through it yourself.)

(b) Use the Python package `qpsolvers` to solve the dual problem for the dual variable ν (following our class notation). See the link below for installation and implementation instructions.

`https://pypi.org/project/qpsolvers/`

(Note: On my 2024 mac, I had some problem installing some of the qp solvers used in the package. The default solver is quadprog and seems to install without difficult. To use it set `solver="quadprog"` in your call to `solve_qp`. However, quadprog requires a positive definite quadratic term. The dual involves the term of the form $\nu^T A \nu$ with A positive semidefinite. You can shift by $A + 10^{-6} I$ to make the term positive definite.)

(c) Use ν to compute a, b and identify the support vectors (i.e. the x_i for which $\nu_i \neq 0$. Plot the data colored by the y_i with the support vectors plotted a different color. On top of the data, plot the SVM hyperplane. Verify that your accuracy is 100%. (There is no need to split into training and test here.)

3. (In this problem, we will develop the theory for kernel, non-separable SVM. To keep things manageable, you don't need to fit the SVM to a dataset. We will do that next week.)

Consider a dataset (x_i, y_i) for $i = 1, 2, \ldots, n$ with $x_i \in \mathbb{R}^p$ and $y_i \in \{-1, 1\}$. Recall the SVM for non-separable data.

$$\min_{a \in \mathbb{R}^p, b \in \mathbb{R}} \|a\|^2 + C \sum_{i=1}^{n} \xi_i$$
$$\text{subject to } y_i(a^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0.$$

We will consider a kernel version of this problem. Let $k(x, y)$ be a reproducing kernel. Define $z_i = k(x, x_i)$ and consider the new dataset (z_i, y_i) and the kernel version of the primal problem,

$$\min_{a \in H, b \in \mathbb{R}} \|a\|^2 + C \sum_{i=1}^{n} \xi_i$$
$$\text{subject to } y_i(a^T z_i + b) \geq 1 - \xi_i, \xi_i \geq 0.$$

Note that $a \in H$ and $z_i \in H$ where H is the RKHS associated with the kernel. Properly, H could be an infinite dimensional space, but for this problem we will treat H as a finite dimensional Euclidean space. The analysis for an infinite dimensional space is essentially the same. So you can take $H = \mathbb{R}^q$.

(a) Derive the dual problem for the kernel problem. (I essentially did this in class.)

(b) For $x \in \mathbb{R}$, show that the SVM predicts $y = \text{sign}(\sum_{i=1}^{n} \nu_i y_i k(x_i, x))$. (Note a feature of the SVM: prediction requires the use of the training data.)

Homework #13

1. In this problem you'll implement spectral clustering on the MNIST dataset.

   (a) Form a knn graph for the MNIST dataset. You may use sklearn to do this. Use your knn graph to from the graph Laplacian. $k = 30$ is a common choice.

   (b) Now using the first ℓ eigenvectors of the graph Laplacian that are not constant (remember here we want the smallest eigenvalues), embed the samples in $\mathbb{R}^\ell$ and apply kmeans with $k = 10$. You may use sklearn's kmeans function. Reuse your code from hw 11 to describe the image distribution in each cluster. What is the best ℓ? Do you do better than using k-means directly on the data?

2. In this problem, you'll provide the details for the Bayes optimal classifier results we discussed in class.

   (a) Define a probability distribution on (x, y) with $x \in \mathbb{R}^2$ and $y \in \{0, 1\}$ as follows. With probability $1/2$, $y = 0$ and $x \sim \mathcal{N}(0, I)$. (This means the pdf of x is given by,

   $$P(x) = \frac{\exp[-\|x\|^2/2]}{2\pi}) \tag{1}$$

   With probability $1/2$, $y = 1$ and $x \sim \mathcal{N}((\mu, 0), I)$. (This means the pdf of x is given by,

   $$P(x) = \frac{\exp[-((x_1 - \mu)^2 + x_2)^2/2]}{2\pi}) \tag{2}$$

   Determine the Bayes optimal classifier $\Phi(x) : \mathbb{R}^2 \to \{0, 1\}$ for this distribution.

   (b) Do **one** of the following.

   - Write a python function to sample (x_i, y_i) for $i = 1, 2, 3, \dots, n$ from this distribution. Set $\mu = 1$. Numerically fit a logistic regression to the data and compare the decision boundary to the Bayes optimal decision boundary. Show that as you increase n, the logistic regression boundary approaches the Bayes optimal boundary. For some value of n, plot the data and the decision boundary. (You may use sklearn to fit the logistic regression. For sampling, you can use scipy's multivariate normal sampler.)

- Let $\ell(\alpha)$ be the log-likelihood of the logistic regression given n samples from the distribution. Show that the solution of $E[\nabla \ell(\alpha)] = 0$ gives a logistic regression decision boundary equal to the Bayes optimal boundary. (Given this result, a law of large numbers arguments shows convergence to the Bayes optimal boundary as $n \to \infty$.)

(c) Now consider a different distribution on (x, y). With probability $1/2$, $y = 0$ and $x \sim \mathcal{N}(0, I)$, as in (a). With probability $1/2$, $y = 1$ and $x \sim \mathcal{N}((10, 0), 5I)$ (This means the pdf of x is given by,

$$P(x) = \frac{\exp[-((x_1 - 10)^2 + x_2)^2/10]}{10\pi}) \tag{3}$$

Show that the decision boundary of the Bayes optimal classifier is not a line.

3. Recall homework 12, problem 3. In this problem we will solve the non-separable SVM problem introduced in that problem. The file `dataset_non_separable.csv` provides non separable data (x_i, y_i) for $i = 1, 2, \ldots, 2000$. $x_i \in \mathbb{R}^2$ and $y_i \in \{-1, 1\}$. For the kernel, use

$$k(x, y) = \exp[-\|x - y\|^2/.01] \tag{4}$$

Split the dataset into training and test datasets, each with 1000 samples. Recall the dual problem from hw 12, problem $3(a)$. For different values of C (the penalty parameter),

- Use qpsolver to compute the dual variable ν using the training dataset.
- Determine the accuracy of the fitted SVM for both the training and test dataset. Recall hw 12, problem $3(b)$ which shows how to predict y given ν.
- Plot the data with the points colored by cell type and then use Python's contour function to show the separating curve in $\mathbb{R}^2$.

Plot the trade-off between training and test accuracy as a function of C. What is the optimal C?

Homework # 14

In this homework, we will use an autoencoder to dimension reduce the MNIST dataest to a 2 dimensional latent space and we'll compare the dimension reduction to what we get using PCA. We'll do the dimension reduction in two ways. Through an autoencoder that uses only the images, but not the accompanying image numbers labels, i.e. $y = 2$ if the image is of a 2. This is an unsupervised approach. And through an encoder with the addition of a prediction layer that does use the accompanying number labels.. This is a supervised approach.

I've attached code solutions for both problems. Feel free to use the code, but use this opportunity to learn pytorch or improve your pytorch coding. Besides the attached code, ChatGPT and Claude (I think Claude is a bit better) are great resources for writing and understanding code in pytorch.

1. Use pytorch to fit an autoencoder to the MNIST data. If X is the MNIST data, rescale to $X/255$ to make all MNIST values lie between 0 and 1; this is just a convenient normalization. From the full MNIST dataset, leave out 5000 samples that will serve to test the autoencoder. In training the autoencoder, use all of the other 55000 samples.

    - The encoding layers should have dimensionality $784, 500, 250, k$ where $k = 2$ is the dimension of the latent space.
    - Use a ReLu between the encoder layers.
    - The decoder layers should have the reverse dimensionaliy.
    - Use a ReLu between the decoders layers, but then apply a sigmoid on the last layer to force the output values to lie between 0 and 1.
    - Train using batches of 100. This means that you should split the training dataset into groups of 100 and each update to the autoencoder should be based on 100 samples. Cycle through the batches to use the full 55000 samples. Each such cycle is referred to as an epoch. Train over several epochs, 10 for example.
    - Train using the mean squared error as the loss function. Code your own loss function, don't use pytorch's.

(a) Recall problem 2 of homework 4 in which we used PCA to dimension reduce MNIST to k dimensions. Here use $k = 2$. Letting x_i be the ith test image and $z_i = e(x_i) \in \mathbb{R}^2$ be the dimension reduction of x_i determined by the encoder, plot the z_i and color according to the number of the images. Do this using the encoder from the autoencoder and from the PCA. Compare the dimension reductions.

(b) Now we'll compare the decoders. Consider the first 10 images in the test dataset, $x_i \in \mathbb{R}^{784}$ for $i = 1, 2, 3, \ldots, 10$. Then $\tilde{x}_i = d(e(x_i)) \in \mathbb{R}^{784}$ where $d()$ and $e()$ are the decoder and encoder respectively. Visualize the $\tilde{x}_i$ by converting to a 28 × 28 matrix and using `imshow` or some other matrix display function:

```
from matplotlib import pyplot as plt

x = x.reshape(28,28)
plt.imshow(x)
```

Do this for the autoencoder and PCA. Which does a better job recovering the images?

2. Now we'll take a supervised approach to dimension reduction. Form a neural net by starting with the encoder architecture of problem 1, but then add a linear and softmax layer to the $k = 2$ dimenionsal output of the encoder:

```
import torch.nn as nn

nn.Linear(k, 10),
nn.Softmax()
```

Letting $f(x)$ be this neural net, $f(x) : \mathbb{R}^{784} \to \mathbb{R}^{10}$ and the soft max gaurantees $\sum_{i=1}^{10} f_i(x) = 1$ where $f_i(x)$ is the ith coordinate of $f(x)$. Letting y be the number shown in the image of sample x, we assume the model $f_i(x) = P(y = i \mid x)$, i.e. the probability that the image is of the number i.

(a) What loss function should be used to train this neural net? Train the neural net with $k = 2$ using the loss function you pick.

(b) Repeat 1a, using the encoder portion of the neural net to dimesion reduce the test dataset. Compare to the plots you generated in 1a.